

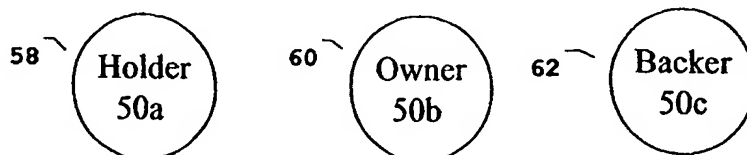


INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

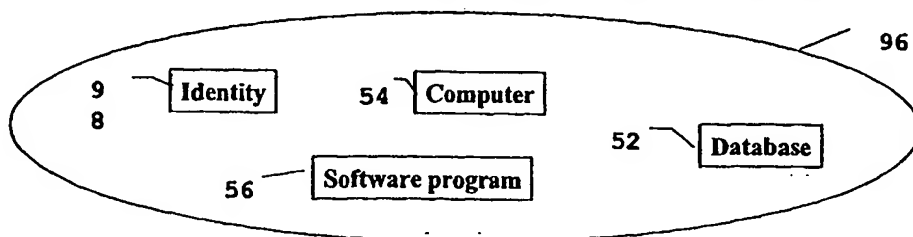
(51) International Patent Classification ⁷ : G06F 17/60, H04K 1/00, H04L 9/00	A1	(11) International Publication Number: WO 00/28452 (43) International Publication Date: 18 May 2000 (18.05.00)
(21) International Application Number: PCT/US99/25853 (22) International Filing Date: 3 November 1999 (03.11.99) (30) Priority Data: 60/107,261 5 November 1998 (05.11.98) US 09/292,291 15 April 1999 (15.04.99) US (71) Applicant (for all designated States except US): SECURE ACCOUNTS LTD. [GB/GB]; 949 Old Ta, The Valley (AI). (72) Inventors; and (75) Inventors/Applicants (for US only): GREEN, Robert [CA/GB]; P.O. Box 931, Shoal Bay (AI). STAMMERS, Jeremy [NZ/GB]; P.O. Box 949, Old Ta (AI). CATE, Vincent [MZ/GB]; P.O. Box 949, Old Ta (AI). HASTINGS, Sean [US/US]; 360 Grand Avenue #105, Oakland, CA 94610 (US). (74) Agents: WEISS, Franklyn et al.; Carr & Ferrell LLP, 2225 East Bayshore Road, Suite 200, Palo Alto, CA 94303 (US).	(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG). Published <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>	

(54) Title: SECURE ARCHITECTURE FOR EXCHANGE EXECUTES DIGITALLY SIGNED CONTRACTS

Parties Play Roles in Exchanges



and Own Resources that Create a Server



(57) Abstract

A secure architecture for value exchanges among parties (58, 60, 62), such that the parties can be widely distributed geographically and can employ a wide range of Accounting Units. It comprises a formalized legal contract in a special format, a public key that identifies each party, a computer for each party (54), a software program (56) for each party that can create, validate, sign, store, encrypt, transmit, and execute contracts, a communication method between the computers, and a database (52) for each party to hold the contracts and Identity information (98) for each party.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

**SECURE ARCHITECTURE FOR EXCHANGE EXECUTES DIGITALLY SIGNED
CONTRACTS**

5

Robert M. Green, Jeremy M. Stammers, Vincent A. Cate,
Sean Hastings

10

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional
Application No. 60/107,261, filed on November 5, 1998 and
entitled "SECURE ARCHITECTURE FOR EXCHANGE EXECUTES
15 DIGITALLY SIGNED CONTRACTS". In addition, this application
claims the benefit of the filing date of U.S. Application
09/292,291, filed on April 15, 1999 with the same title.
Both applications are herein incorporated by reference.

BACKGROUND OF THE INVENTION

20 **1. Field of the Invention**

This invention relates to the use of computers,
networks, and encryption algorithms to manage and secure
exchanges of value, and more particularly to a computer
architecture and system for performing and managing value
25 exchanges that operate on a wide range of inexpensive
computer systems.

2. Description of the Prior Art

Until the early 1950s, transactions to exchange value, such as invoicing, payments, wire transfers, market trades, conversions from one value unit to another, were handled manually by specialized paperwork and human processing. This manual processing was slow and error prone. As computers were applied to these activities, specific computer systems automated particular types of value transactions, usually book entry systems. Often the manual processes were copied as is into computer form and automated with the same restrictions, assumptions, and context as the manual process.

With the invention of encryption, especially public key encryption, some value transactions were made more secure from intrusion and interference. However, these prior computer systems, protocols, and architectures have the following disadvantages:

- they are specialized for certain transactions only and often for certain geographic areas only;
- some require specialized hardware, such as Smart Cards;
- some require private communications networks and will not work on public networks, such as SWIFT;
- some require continuous, two way communication to validate and record transactions;
- some are closed to new entrants (i.e., they work only with existing organizations, such as protocols for authorizing credit and debit cards, which only work with banks; the architectures have elements of the existing organizations and businesses built into them;

- some are closed systems which do not lend themselves to integration into customized applications;
- they are not secure (i.e., they do not use strong encryption);
- 5 • they cannot handle multiple accounting units in the same transactions;
- they assume a hierarchical, asymmetrical relationship between parties in a value exchange (that is, someone is always the bank and who cannot also be the merchant or customer in the same transaction); in computer
- 10 system terms, they distinguish strongly between the client and server - a party is either one or the other but not both;
- they are explicitly based on a specific regulatory commissions and limited to that commissions' view of value transactions.
- 15 • they can only handle an arbitrarily small number of parties in one transaction, often only three or four (one customer, one merchant, one bank and one credit card organization);
- 20 • they are inherently structured to be used in a limited number of legal jurisdictions;
- they can only function on-line and have no ability to complete an exchange off-line;
- 25 • they assume a fixed degree of trust between parties, often specified by an organization at the top of a hierarchy that created the protocol; if you do not accept your place in the hierarchy and their rules on how much to trust parties, you cannot use the protocol;

- some, such as the ANSI EDI standards, are document presentation protocols, without digital signatures, non-repudiation and bookkeeping.

5 "Value exchanges" in the abstract sense are a part of payments, invoicing, bartering, market making, wire transfers, stock trades, issuance and clearing of checks, and currency conversions. The existing prior art performs some, but not all of these value exchanges, or only performs
10 the communication needed prior to the exchange and not the exchange itself.

Other Efforts: There are three basic approaches to electronic payments and/or value exchanges: software-coin
15 based, account-based (there are banks who know who the money belongs to - the present invention falls in this category), secure-hardware based (Mondex- the money is contained in the card).

- 20
- Software-coin approach
 - eCash (tm) from Digicash.
 - Account approach
 - Credit Card Based Systems
 - Netscape's SSL Protocol for transmitting your
25 credit card.
 - First Virtual allows you to authorize use of your credit card without giving out the actual card number.
 - CyberCash(tm). Attaches a public key to a
30 credit card.

- SET - Secure Electronic Transaction from MasterCard, Visa, American Express etc
- Other Systems:
 - 5 • Open Trading Protocol is a method of making incompatible payment systems interoperable. It sits on top of other payment systems such as Mondex and is supported by a consortium of payment vendors.
 - 10 • Open Financial Exchange is a standard for formatting requests and responses in a markup language, but does not specify how the requests are processed; security via SSL is optional;
 - 15 • JEPI - Joint Electronic Payment Initiative from CommerceNet and W3C (the World Wide Web Consortium)
 - The SAXAS™ System (the invention described herein)
 - Secure-Hardware approach
 - 20 • ATM Systems.
 - Smart Cards.
 - Mondex Electronic Cash. Embeds the money and the secret that protects it in a special smart card.
 - E-Check - Electronic Checkbook on a smart card
 - 25 from the US based Financial Services Technology Consortium and FSTC members; Federal Reserve Bank, NationsBank, Bank of Boston, Huntington Bancshares, IBM and Sun Microsystems and others.
 - EMV - Debit/credit cards using chip technology
 - 30 from Europay, MasterCard and Visa.

A number of these efforts are worth describing in more detail, in order to create a comparison between them and the present invention, the SAXAS system.

5

Mondex Electronic Cash extends the ATM model by making smart cards where the money and the secret that protects it are embedded in a special card. The security of the money is dependent upon special hardware that is difficult to break
10 into. To spend the money, you must insert the card into an ATM or special reader. If you cannot read the internals of the card, you cannot duplicate it. If you could duplicate the card, you could steal the "money". The money belongs to the physical card; if you lose the card, you lose the
15 money. If someone steals the card, they may not be able to spend the money, if the card has a PIN, but you have still lost it. There is no output from the card except to a card reader such as the ATM. If a phony ATM on the street was taking all your electronic money, but telling you it was
20 not, you would have no way of knowing until you tried to use the card again later at a real ATM. Mondex will probably be used mainly for small amounts, with no accounting record.

This model of electronic money is patterned after real
25 physical money. The value is recorded in a unique physical configuration that is difficult to counterfeit and can be used anonymously.

CyberCash, on the other hand, puts a public key around
30 an existing credit card number in order to allow merchants to do real-time credit card authorizations. CyberCash is converting its system to use SET, which is a version of this

approach created by the credit card organizations themselves.

The present invention, the SAXAS system, can easily
5 emulate the wrapping of a credit card in a public key: in
the agreement field of a SAXAS contract, you authorize
another party to charge your credit card, and then you
digitally sign the contract. But SAXAS can do much more:
handle multiple accounting units, match up complementary
10 conversion orders, and transfer values without a credit card
being involved.

eCash™, a product of Digicash, offers software-based
electronic coins that can be spent at eCash merchants and
15 verified at a central point or bank. Since eCash "coins" can
be digitally duplicated, the first person to the bank gets
the money. They can only be used once and each has a unique
serial number. There is no special hardware required, since
each coin is just a string of bits that equals the right to
20 a specific amount of money.

eCash assumes a central issuing authority, modeled
after existing currency issued by governments and deposited
at banks. When you transfer eCash to another party, the bank
25 must play a part. That is because eCash coins can be
duplicated and can only be spent once. Therefore, eCash must
have a central database that lists all the serial numbers of
coins issued and which have been spent so far. SAXAS allows
accounting unit balances to move from party to party without
30 the issuing bank being aware of the transfer. ECash uses a
Blind Signature and cannot trace its own cash.

The SAXAS system can emulate the eCash capability by creating a one-time identity for a SAXAS contract, then including the private key as well as the public key within the contract. Anyone receiving the contract could sign it, execute it, then do an Owner change to move the amount from the one-time account to his or her own account.

First Virtual is an Internet payment system that allows you to circumvent sending your credit card over the Internet. You set up an account with them, register your credit card over the phone, and type in your password when prompted at sites charging using this system. You have to wait 60 days for your money, and it is one of the most barrier free methods of receiving payment over the Internet, you do not even need a merchant credit card account to accept credit cards. You do need to set up some scripts on a working First Virtual server.

Open Trading Protocol (OTP) is a consortium that is attempting to create "an open and interoperable standard for purchasing goods and services on the Internet." The model behind OTP is the existing business of purchasing goods and services. Existing types of participants in this model are explicitly identified in the model (i.e., merchant, consumer, deliverer, customer care agent, bank). Interoperable is a key word here. This builds on top of existing payment systems to allow them to communicate. It does not actually do the payment or the exchange itself as the present invention, SAXAS, does. OTP does not assist you to create and manage your own accounting units, but if you did create one with SAXAS it might help you inter-relate it to other payment schemes. OTP derives from traditional

paperwork methods of exchanging goods and services, and is a protocol for presenting paperwork; it builds on top of and assumes some payment methods. It could build on top of SAXAS.

- 5 OTP assumes backend systems for payments and record keeping. Account balances and inquiries for information on previous transactions are not available.

10 Open Financial Exchange (OFX) is a markup language for creating requests and formatting responses. It specifically tags for USA-based financial attributes such as 401K Plans, etc. Security via SSL is optional. OFX does not specify how exchanges are processed or does it perform exchanges itself. OFX looks like an extended version of HTML (HyperText Markup
15 Language) with tags for specific financial structures. It imposes a structure on the messages that is mapped on the specific types of financial institutions that are the intended audience of the protocol. That is, it is not a generalized method for doing any value exchange.

20

 Instead it is a formatting standard for bank transactions, wire transfers, credit card transactions, payments, and investments, each of which is specifically identified in the protocol. For example, only certain
25 country codes are recognized and they are listed in the protocol definition. It is hierarchical in that clients are recognized by password and servers are recognized by certificate. In SAXAS, by comparison, parties are symmetrical and the mechanism is general-purpose.

30

 In OFX, the servers are identified by X.509 certificates. If the client and server do not share a common

Certificate Authority (CA), the client cannot validate the server's certificate. Therefore, OFX specifies which CAs are to be trusted by OFX clients.

5 There are currently two main Certificate Authorities that have achieved market acceptance: Verisign and Thawte. They perform due diligence on a person or firm or web page or email address, then "certify" that the public key actually came from the entity in question, and generates an
10 X509 certificate that a party can supply to strangers to vouch for them.

 SAXAS allows all parties to identify themselves with X.509 certificates, but such certificates are optional in
15 SAXAS and it is up to a particular SAXAS server to decide which certificates it will require. X509 certificates are not essential to performance of the invention set forth in this application. In the SAXAS system being described herein, the parties to a contract may require a particular
20 type of certificate from the parties, or they might not, depending on the situation.

Prior Art Patents

25 The prior art includes Payne et al, U.S. Patent No. 5,715,314, issued Feb. 3, 1998, and discloses:

 A network-based sales system includes at least one buyer computer for operation by a user desiring to buy
30 a product, at least one merchant computer, and at least one payment computer. The buyer computer, the merchant computer, and the payment computer are interconnected

by a computer network. The buyer computer is programmed to receive a user request for purchasing a product, and to cause a payment message to be sent to the payment computer that comprises a product identifier
5 identifying the product. The payment computer is programmed to receive the payment message, to cause an access message to be created that comprises the product identifier and an access message authenticator based on a cryptographic key, and to cause the access message to
10 be sent to the merchant computer. The merchant computer is programmed to receive the access message, to verify the access message authenticator to ensure that the access message authenticator was created using the cryptographic key, and to cause the product to be sent
15 to the user desiring to buy the product.

Moreau, U.S. Patent No. 5,590,196, issued Dec. 31, 1996, discloses:

20 Electronic funds transfer processes are being put into place to replace the paper based check clearing process. Although ubiquitous in the business environment, facsimile transmission technology has not been used for electronic transfer of funds. Fraud
25 prevention and uncertainties in the legal status of a facsimile transmission are among the impediments for electronic funds transfer with facsimile. The method for transferring funds from a payer to a payee comprises the steps of preparing a payment form
30 including information for identifying an amount to be transferred, a bank of the payee and an account number of the payee, receiving and verifying a security code

at an encryption
unit to authorize a transmission including an
encryption, preparing a facsimile transmission device
to send an image of the payment form, connecting the
5 facsimile device through the encryption unit over a
communication line to a payment service provider,
receiving at the payment service provider the
transmission including an encryption, and sending a
confirmation message to the facsimile device that the
10 transmission has been correctly received, decrypting
the encryption at the payment service provider,
determining whether the encryption was authentically
generated by the payer, extracting the identifying
information from the facsimile transmission, and
15 generating an electronic funds transfer request based
on the identifying information provided that the
encryption is determined to be authentic.

Rosen, U.S. Patent No. 5,671,280, issued Sept. 23, 1997,
20 discloses:

A system for electronic commercial payment is provided
having a customer trusted agent associated with a first
money module, a merchant trusted agent that establishes
25 a first cryptographically secure session with the
customer trusted agent and associated with a second
money module. Where the money modules establish a
second cryptographically secure session. The customer
trusted agent provides remittance advice information to
30 the merchant trusted agent, and the merchant trusted
agent provides a commercial payment ticket to the
customer trusted agent. Upon receiving said commercial

payment ticket, the customer trusted agent initiates a transfer of electronic money from the first money module to the second money module.

5 Chasek, U.S. Patent No. 5,420,405, issued May 30, 1995, discloses:

10 This invention describes a combination of methods and apparatus that creates electronic money for personal transactions which integrates the functions of cash, checks and credit cards with constant surveillance against fraud. This money can also serve as an international medium-of-exchange, and support automated sales tax collections and payment. This money's support
15 system is comprised of personal terminals, vendor terminals, an electronic banking sub-system, and homebase terminals. Such a system, if widely used, would increase commercial and personal productivity, provide better security against fraud and
20 counterfeiting, facilitate the automation of operations that involve currency, and sharply diminish the flood of paper that threatens to inundate the present system.

Micali, U.S. Patent No. 5,629,982, issued May 13, 1997,
25 discloses:

A number of electronic communications methods are described involving a first and second party (i.e., sender and recipient), with assistance from at least a
30 trusted party, enabling electronic transactions in which the first party has a message for the second party. The first party, the second party and the

trusted party undertake an exchange of transmissions,
such that if all transmissions reach their destinations
the second party only receives the message if the first
party receives at least one receipt. Preferably, the
5 identity of the first party is temporarily withheld
from the second party during the transaction. At least
one receipt received to the first party enables the
first party to prove the content of the message
received by the second party.

10 Elgamal, U.S. Patent No. 5,671,279, issued Sept. 23, 1997
discloses:

15 A courier electronic payment system provides customers,
merchants, and banks with a secure mechanism for using
a public network as a platform for credit card payment
services. The system governs the relationship between a
Customer, Merchant, and Acquirer Gateway to perform
credit card purchases over such networks as the
20 Internet. The system uses a secure connection to
simplify the problem of Internet-based financial
transactions in accordance with an electronic payment
protocol that secures credit card payments and
certifies infrastructure that is required to enable all
25 of the parties to participate in
the electronic commerce, as well as to provide the
necessary formats and interfaces between the different
modules and systems.

30 Gifford, U.S. Patent 5,724,424, issued Mar 3, 1998,
discloses:

Merchant computers on the network maintain databases of digital advertisements that are accessed by buyer computers. In response to user inquiries, buyer computers retrieve and display digital advertisements from merchant computers. A digital advertisement can further include a program that is interpreted by a buyer's computer. The buyer computers include a means for a user to purchase the product described by a digital advertisement. If a user has not specified a means of payment at the time of purchase, it can be requested after a purchase transaction is initiated. A network payment system performs payment order authorization in a network with untrusted switching, transmission, and host components. Payment orders are backed by accounts in an external financial system network, and the payment system obtains account authorizations from this external network in real-time. Payment orders are signed with authenticators that can be based on any combination of a secret function of the payment order parameters, a single-use transaction identifier, or a specified network address.

Objects and Advantages

Accordingly, several objects and advantages of this invention are:

- to be applicable to a wider range of value transactions, including invoicing, payments, purchase, sell, barter, wire transfers, market trades, conversions, exchanges, stock markets, purchase and delivery of intellectual property such as software,

time limited exchange offers, deposits, withdrawals,
and loans;

- to be secure enough that transactions are tamper proof and non-forgable.
- 5 • to operate on off-the-shelf, standard, inexpensive computer hardware and software as well as more expensive proprietary systems such as mainframes.
- to use public networks such as the Internet while still maintaining privacy and security.
- 10 • to enable any organization to create their own Accounting Units to facilitate general or limited value exchanges; examples include, but are not limited to, banks that require a way to manage customer balances, Frequent Flyer Miles for airlines, gambling tokens for
- 15 casinos, cooperative marketing "coupons" that allow a customer to purchase related products from other vendors, etc.
- to facilitate the creation of public markets in these Accounting Units by matching buyers and sellers via
- 20 automatic software.
- to integrate easily with external applications such as Internet Shopping or legacy accounting systems.
- to be independent of any specific regulatory environment or jurisdiction and applicable to multiple
- 25 jurisdictions.
- to integrate any existing legal agreements for value exchange into a computerized system for signing, auditing, and accounting for them.
- to maintain account balances in a way that fraud can be
- 30 avoided while privacy is still maintained.

- to be decentralized over a wide range of servers and services so that there is not a single point of failure that can bring down the entire system, either through physical failure or human intervention.
- 5 • to have server functionality independent of a fixed network location (that is, if a server is disabled, the party running that server can reappear at another location and be recognized and in a known state relative to value transactions).

10

Other objects and advantages include:

- to process exchanges that are not valid until a certain date and time and/or which expire on a certain date and time.
- 15 • to process partial exchanges and multiple exchanges over a specified period of time at a specified limit price (example: between June 1 and June 6, buy 1000 shares of Secure Accounts Ltd. at a maximum of 100 Unibank credits per share).
- 20 • to process exchanges involving unlimited numbers of parties as one exchange (i.e., either all exchanges occur or none); this invention makes it possible for a party to act as escrow agent for an arbitrarily large number of parties to a single transaction.
- 25 • to allow competing Accounting Units and exchange services to use the same architecture
- to allow a complete audit trail of all exchanges, enabling account balances to be regenerated if necessary due to system failure or dispute.
- 30 • to provide non-repudiatable contracts for value exchanges.

- to allow a selectable degree of confidence that you know who you are dealing with (the architecture allows any party providing a trusted service to require a specific type of digital signature from the parties that he deals with).
- to provide interfaces for extensibility of the architecture by other parties.
- to allow private branding of value transactions by other parties who will integrate this invention as a foundation for their specific exchange applications.
- to provide simpler methods for propagation to the parties the final transaction documents, including delivery by email or even physical mail; the underlying delivery model is neutral to the method employed - it need only be a one-way delivery.
- to allow any party to nominate any trusted third party to be the holder of account balances and executor of value exchanges.
- the ability to do offline transactions, such as "bank cheques", that will facilitate migration of store and forward exchange systems to fully online systems.
- to be technology neutral.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the invention, as well as other objects and further features thereof, reference may be had to the following detailed description of the invention in conjunction with the drawings wherein:

Fig. 1 shows the relationship of the parties and resources used in a value exchange;

Fig. 2 shows a public key that can define and identify a party and digital signature which can verify that a party
5 agrees to the contract;

Fig. 3 shows an X.509 certificate that is an alternate way to define and identify a party;

Fig. 4 shows an account defined by a unique triplet of Holder, Owner and Backer, plus a balance;

10 Fig. 5 shows several different types of accounts;

Fig. 6 shows the accounts created for one backer, two holders and three owners;

Fig. 7 shows the accounts created when a holder is also an owner of units;

15 Fig. 8 shows the structure of a value exchange contract;

Fig. 9 shows the three types of contracts;

Fig. 10 shows a Single-Party Owner Change Contract with an agreement;

20 Fig. 11 shows the structure of a multi-party Owner Change contract;

Fig. 12 shows Holder Change Contract, from Original Holder to Backer;

25 Fig. 13 shows Holder Change Contract, from Backer to new Holder;

Fig. 14 shows the structure of a Backer Change contract;

Fig. 15 the same accounts in Fig. 6, but with Friendly Names instead of element numbers;

30 Fig. 16 shows the execution logic for an Owner Change contract;

Fig. 17 shows the execution logic for a Holder Change contract;

Fig. 18 shows the execution logic for a new Backer Change clause;

5 Fig. 19a and 19b show examples of matching Backer change clauses;

Fig. 20 shows a queue of waiting Backer change clauses matched to a new backer change clause; and

10 Fig. 21 shows the queue of Backer change clauses after matching a new clause.

SUMMARY OF THE INVENTION

This invention involves a secure architecture for value
15 exchanges among parties, such that the parties can be widely distributed geographically and can employ a wide range of Accounting Units. It comprises a formalized legal contract in a special format, a public key that identifies each party, a computer for each party, a software program for
20 each party that can create, validate, sign, store, encrypt, transmit, and execute contracts, a communication method between the computers, and a database for each party to hold the contracts and Identity information for each party.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

5 This invention discloses an architecture for performing and managing value exchanges, with an initial embodiment in the Secure Accounts eXchange Arbitration System (also called SAXAS™) that operates on a wide range of inexpensive computer systems.

10 Fig. 1 shows the main components of the SAXAS architecture of value exchange. An actor in an exchange is called a Party 50. There are three roles that a party can play in any given exchange: the Holder 58, the Owner 60, and the Backer 62. A party may play more than one role in a given exchange and a
15 party's role may change for each exchange. The architecture allows any party to be a Backer, a Holder or an Owner, or all three.

 In order for a party to exist and take part in
20 exchanges, it must have a unique Identity 98, a computer 54, a database 52, and a copy of the SAXAS software program 56.

 The Identity 98 of a party consists of a unique Public Key 68, Fig. 2, plus any information that the party wishes
25 to make public. The Identity is signed by the party. This digital signature ensures that the identity information does go with the public key. The public key is also used to encrypt communications with the party, ensuring that only the intended party can read the message.

30

The information in the Identity includes a Notify Method which specifies how the SAXAS software can communicate with the party. This field will contain a URL (Universal Resource Locator), which can specify a Internet address or an email address. The combination of a Public Key and a Notify Method creates a secure channel for notifying parties of SAXAS events that are of interest to them.

The identify information may also include a preferred friendly name, telephone number, web page address, and details on any SAXAS services that the party offers.

The Identity may also include a list of attached documents, including, but not limited to, X.509 certificates and graphic logo images.

The computer 54 could comprise a personal computer (PC) running Windows® 95 or Windows® 98 with a minimum of 32MB of memory and at least 5MB of free disk space to manage the database, code and installation requirements. If the computer 54 is a personal computer (PC) running Windows NT Workstation or Server software, then it requires a minimum of 64MB of memory and 5MB of free hard disk space. When the JAVA program is utilized, then versions JRE 1.2, JDK 1.1.6, JIT 1.1.7 are involved and available on the Internet at <http://www.javasoft.com>. The database 52 components could include a database engine compatible with JDBC, such as SQL Anywhere 5.0 from Sybase or Access from Microsoft. Also required is a JDBC driver and an ODBC drive for this database. The SAXAS software required would include an Installshield or self-extracting

zip file containing the SAXAS software, JRE 1.2, JDK 1.1.6, JIT 1.1.7. The communications requirement includes a TCP/IP connection to other SAXAS sites, such as provided by an Internet Service Provider through a modem, or
5 indirectly through a LAN using Network Interface Cards and then through an Internet Connection.

The combination of resources belonging to a party (98, 54, 52, and 56) creates a Server 96 for the Party, which is
10 capable of managing recording the exchanges undertaken by the Party. It is possible for a single physical server to act as the logical server for a number of parties; the only requirement is that these parties share the same Transport public key for decoding incoming contracts. The function and
15 use of these resources is explained further in the drawings and description that follow.

The Backer 62 is the party that issues or "backs" an Accounting Unit, which can also be considered as an
20 electronic currency or virtual currency. The backer guarantees the accounting unit value by providing exchange to a specified Real World value. The backer must also act as holder and support a market for the accounting unit, matching buy and sell orders. The backer may also act as
25 holder for other accounting units and provide exchanges to and from them to his accounting unit. The backer may also act as the owner of accounts with other holders.

The Backer can be a government that backs legal tender, a bank that backs receipts for legal tender or for shares in a
30 Money Market Mutual fund, a casino that issues gambling tokens, an airline that issues frequent flyer miles, a corporation that issues phone cards, or ownership shares, or

coupons that can be redeemed at cooperating companies, a lawyer who issues units redeemable as hours of his time, or any other value that someone will promise to redeem under certain conditions with some real world value. The

5 architecture does not ensure in any way that the backers are reliable or can be trusted to keep their promises. What the architecture does do is create an open market in their accounting units such that the exchange value of the accounting units can be marked down if the backer is
10 unreliable. Therefore, what is backing an accounting unit is actually the reputation of the backer.

An Accounting Unit is equivalent to the public key that identifies the backer 62. A specific person or organization
15 could of course have several public keys and act as backer for several Accounting Units.

The Holder 58 is typically a bank, escrow agent or broker. The Holder keeps amounts for other Owners in the
20 Accounting Units of one or more Backers. The holder may provide various exchange services that are publicly defined in the architecture and available for computerized execution. The holder must be the Owner of an account with each Backer whose Accounting Unit it holds. The holder may
25 also be an Owner of accounts with other holders and may also be the backer of an accounting unit. The holder may provide whatever degree of anonymity or lack thereof that he decides is appropriate to his business. The Holder enforces knowledge of parties by requiring specific types of
30 Certificates 74 in a party's Identity 98.

The Owner 60 is the signatory for one or more accounting unit balances at one or more holders. The owner must sign for any transfers from those balances. The owner may also be a holder or a backer. The owner identity may be
5 a temporary identity that is used only one-time. The degree of anonymity allowed to an owner is determined by the holders who accept his balances.

Fig. 2 shows the format of a public key 68 and a
10 digital signature 114. A public key is a sequence of unforgeable characters that can uniquely and securely identify the party who holds the corresponding private key (see U.S. Patent No. 4,200,770, for the Diffie-Hellman public key cryptosystem and U.S. Patent No. 4,405,829 for
15 RSA public key cryptosystem). In the SAXAS architecture, each party is uniquely defined by a public key. The private key is used to sign SAXAS contracts and changes to contracts that are specific to the signing party. The result is a digital signature 114, like the one shown in Fig. 2. The
20 public key is used to verify authorizations by the party, ensuring that only the true party can authorize withdrawals of value from that account or exchanges to other Accounting Units. So, the official name and identity of a Party, whether acting as Holder, Owner or Backer, is the public
25 key.

Fig. 3 shows another form of party identity: the X.509 digital certificate. This is a public key plus information about the real world identity of the party, all signed and verified by a certifying authority. A party who acts as a
30 holder can require that the parties it opens accounts for have valid certificates from specified authorities. In this

way the holder can enforce the degree of anonymity it wishes to offer on accounts.

Fig. 4 shows the structure of an account 92 in the SAXAS architecture. An account is uniquely defined by the public key identities of a Holder 58, Owner 60, and Backer 62, and also contains an account balance 94 in the Backer's accounting unit. The same party can appear as one, two or three of the Holder, Owner or Backer parties. Balances can be negative as in double-entry bookkeeping systems. The specific account diagrammed in Fig. 4 is interpreted as "Holder 50a owes Owner 50b the sum of 100 units of Backer 50c accounting units". Transferring a balance to a non-existent account automatically creates a new account for the Owner, unless this feature is turned off by the Holder. The architecture makes it easy for the Holder to charge a fee for maintaining account records and the Holder should charge enough to keep account records indefinitely.

Fig. 5 introduces the different types of accounts that can exist in the SAXAS architecture. When the Holder and the Owner are not the same party, a positive balance indicates that the Holder owes the Owner the balance amount, and a negative balance indicates that the Owner owes the Holder the absolute value of the balance. When the Holder equals the Owner, the account balance indicates the total units owned by the Holder on his own behalf (this amount is zero or positive).

The following account balances from Fig. 5 are complementary and compatible.

- account 92a has a positive balance and means that Holder 50a owes 100 units of 50c to owner 50b.
- account 92b has a negative balance and means that party 50b is owed 100 units of 50c by party 50a. When account 92a exists for party 50a, then account 92b always exists for party 50b. Account 92b is held by party 50b as a claim against party 50a. Notice that the same party can be both holder and owner in different accounts.
- account 92c records the total units issued to holder 50a by backer 50c. Since the balance is negative, the holder has a claim against the owner, who is also the backer.
- account 92d has the Holder and the Owner the same party and a positive balance; this means that party 50b has a total of 100 units of 50c that are held for it by one or more holders. Account 92b shows that the 100 units are actually held by party 50a.

An account resides on the server 96, Fig. 1, belonging to the party 50 who is the Holder specified in the account triplet. Therefore, in Fig. 5, accounts 92a and 92c reside on the server of party 50a, while accounts 92b and 92d reside on the server of party 50b.

Fig. 5 also shows how accounts can be balanced against each other, both on the same server and on different servers.

- Account 92a balances against 92b (that is, party 50a owes 50b 100 units, and party 50b is owed 100 units by 52a).

- Account 92c balances against 92a (i.e., the total 50c units issued to 50a is the sum of the units held for owners such as 50b).

5 Fig. 6 shows the accounts that would be created to handle one backer, two holders, and three owners:

- When all three parts of the triplet are the same party (Holder equals Owner equals Backer), the balance
10 records the total units issued by the backer to all parties and is negative. In Fig. 6 account 92k shows that backer 50c has issued 50 units in total.
- Accounts 92k-92m reside on the 50c server and record
15 units backed by this party. Notice that accounts 92l-92m balance to account 92k, which records the total obligation. Account 92l records the 20 units issued to the first holder 50a and 92m records the 30 units issued to the second holder 50b. Notice that the backer
20 does not keep track of which party actually owns the units, only who holds them.
- Accounts 92n-92p reside on the server of holder 50a and
25 balance to zero. Account 92n is the complement of account 92l on the backer server and records the 20 units received by this holder. Accounts 92o-92p record what happened to those 20 units: 5 units are held for owner 50d and 15 units for 50e.
- Accounts 92q-92s reside on the server of the second
30 holder 50b and balance to zero. Account 92q is the

complement of account 92m on the backer server and records the 30 units received by this holder. Account 92r-92s record what happened to those 30 units: 20 are held for owner 50d and 10 are held for owner 50f.

5 Notice that party 50d has units held by both holders.

- Accounts 92t-92v reside on the server of owner 50d and balance to zero. Account 92t records the 25 units total backed by 50c that belong to this owner. Account 92u
10 records that five of the units are held by holder 50a, and account 92v records that the other 20 are held by holder 50b. Notice that account 92v is the complement of 92r and balances it. And account 92u is the
15 complement of 92o and balances it.

- Accounts 92w-92x reside on the server of owner 50e and balance to zero. Account 92w records the total units of 50c received by this owner and account 92x balances to
20 account 92p on the holder server.

- Accounts 92y-92z reside on the server of owner 50f and balance to zero. Account 92y records the total units of 50c received by this owner and account 92z shows that
25 the units are held by holder 50b (see account 92s).

Notice that the architecture can balance the sum of all the units received by owners to the total units issued by the backers, irrespective of which holder is holding them (accounts 92y, 92w and 92t total 50 units, which balances to
30 the -50 balance of account 92k on the backer's server).

Fig. 7 shows how accounts are balanced when a Holder 58 is also the Owner 60 of some accounting units. In this example, backer 50c has issued 50 units to Holder 50a, of which 50a holds 10 for his own account and 40 for Owner 50b.

5 This is represented by the following accounts:

- 10 • Accounts 92a-92b reside on the server of backer 50c and balance to zero. Account 92a records the total units that backer 50c has issued to all parties and account 92b records the units issued to Holder 50a. In this case they are the same, because only one holder is assumed. Otherwise, account 92a would balance to the sum total of the accounts recording units issued to any holder.
- 15 • Accounts 92c-92e reside on the server of Holder 50a and balance to zero. Account 92c records the total units that backer 50c has issued to the holder 50a, including both those held for owner 50b and those owned by the holder himself. Account 92d records the units actually
20 owned by the Holder 50a himself (the Holder party equals the Owner party). Account 92e records the units held on behalf of Owner 50b.
- 25 • Notice that account 92b on the 50c backer server also balances to account 92c on the holder server. That is because the units issued to 50c are always equal to the units received by 50c.
- 30 • Accounts 92f-92g reside on the server of owner 50b and balance to zero. Account 92f records the total units backed by 50c that the owner 50b has at all holders

(only one in this case) and account 92g records the units held for 50b at holder 92a.

- Notice that account 92e on the holder 50a server
5 balances to 92g on the owner 50b server. That will
always be the case when the Holder and Owner parties
are reversed in the account triplet (i.e., what you
have sent to be held is equal to the amount that the
holder owes you).

10

Fig. 8 shows the structure of a value exchange contract
66. In this architecture, all exchanges of value are enacted
in response to signed contracts. Each of the parties who
appears as a source in the Clause section 82 must sign the
15 top four sections (76, which comprises Header 78, Parties
80, Clauses 82, Agreement 84) of the contract in the fifth
section (Signatures 86). The contracts are signed by the
parties using their Identity (98, either public key 68 or
digital certificate 74), Fig. 1, thus ensuring that everyone
20 agrees to the same contract, then executed by the server.
Signatures do not oversign each other; parties sign only the
signable portion of the contract 76 and the Memo, Sig-Time
and Status specific to that party in the Signatures section.
Only a party listed in the party section 80 of the contract
25 is allowed to sign in the Signatures section 86, thus
allowing verification of the signature using the public key.

The contract resides in the data storage of the SAXAS
server, but can be generated in various formats for external
30 use: display (HTML-format document (HyperText Markup
Language), enabling it to be rendered via a web browser),

edit (also HTML), transmission to another server (as a serialized object), and printing. The format that is signed is a text version of the contract (not the HTML), which the software can display for user inspection and print for
5 submission to a legal authority or arbitrator.

Not all components of the contract are relevant to all types of contracts. The Agreement 84 is empty except in an owner change contract (see Fig. 9b). The "When Contract Becomes Valid" field of the Header 78 is only relevant to
10 backer change contracts (see Fig. 9c); other contracts are valid as soon as they are signed by all parties. The "When Contract Expires" field is not relevant for holder change changes (see Fig. 9a) since they are complete as soon as execution starts.

15

Fig. 9 shows the three most common types of contracts in the architecture. There are different types of contracts with different restrictions on them in order to ensure that
20 no account balances are ever lost or in an ambiguous state. Since an Account 92, Fig. 4, is defined by a triplet of Holder 58, Owner 60 and Backer 62, moving an amount from one account balance to another can be looked at as changing the associated Holder, Owner and/or Backer value.

25

Each contract specifies an Executor 64 in the Header section 78, Fig. 8. This party will also act as the Holder party 58 in all clauses except the destination account of Holder clauses 106. Each contract, regardless of type, may
30 have an optional Fee Clause 90 that is the first clause in the Clause section 82. The fee clause 64 extracts a fee for executing the contract. If it exists, it does an Owner

Change on some amount of some accounting unit from that of the party paying the fee to that of the Executor. The accounting unit of the Fee Clause need not be the same as the accounting unit used in the other clauses. Therefore,
5 there can always be one extra Backer party to the contract if the Fee clause Backer differs from the others.

There are specialized contracts in SAXAS, such as Name Service contracts with supply and request identity
10 information, but these are all variations on the Owner Change contract, with details of the special functions stored in the agreement field.

In subsequent clauses, a contract will only modify one
15 of the values on an account: Holder, Owner or Backer. It is theoretically possible to change more than one attribute of an account in a single contract (such as moving an amount to a new owner and changing the accounting unit at the same time), but in such cases it is more difficult to ensure that
20 the contract executes completely or is not executed at all (i.e., that no amounts are left in limbo). Therefore, the first embodiment of SAXAS does not support that feature, in order to ensure reliability and easy auditing of the system, although other embodiments may allow this (see Alternate
25 Embodiments).

It is the type of changes performed in the clauses section 82 that determines the type of contract and leads to the three types:

- 30 • Fig. 9a: Holder change contract 98 is like a wire transfer; it moves a balance from one Holder to

another, perhaps to then take advantage of a conversion service offered by that Holder, but it does not change the Owner or the Backer (accounting unit). Either the source or destination Holder must be the Backer, so it takes two contracts to move from one non-Backer Holder to another non-Backer Holder. The parties to the contract will be one Owner 60, the Backer 62 and one Party who is the source or destination Holder 58. There can only be one Holder clause 106 in the Clauses section 90. The executor 64 will be the party originally holding the amount, not the destination party. The Agreement section 84 is empty. The Signatures section 86 contains only the signature of the Owner, the Holder and the Backer.

- Fig. 9b: Owner change contract 100 is like a check or an invoice, depending upon the author. When an invoice is signed, it becomes a check. The amount stays on the same Holder's server and in the same accounting unit, but has a new owner. Parties to the contract include at most one Holder 58 and at least one Backer 62 and at least two Owners 60, but it is permissible to include unlimited owners and backers for unlimited owner clauses 108. Either all the clauses are executed or none. The Agreement section 84 may contain one or more real world agreements that becomes part of the contract. The Agreement is open-ended and can contain multiple agreements and objects. Any digital object can be included in the Agreement by converting it into ASCII-armored format using Radix 64 (expanding groups of 3 binary 8-bit bytes into 4 printable ASCII

characters). The Signature section 86 must contain the signatures of all owners who appear in a source account of any clause.

- 5 • Fig. 9c: Backer change contract 102 is like a real
world currency exchange. The server matches up the
request to change Backer with other contracts which go
the opposite way and that have compatible prices. The
parties to the contract include exactly one holder 58,
10 exactly two backers 62 and exactly one owner 60,
although it is possible that a single party could play
multiple roles in the contract. There can only be one
Backer clause 110 in the Clauses section 90. The
Agreement section 84 is empty. The Signatures section
15 86 must contain the signature of the owner 60.

By creating several contracts and executing them in
sequence, it is possible to get an amount from any account
to the account of any Owner residing at any Holder in the
20 Accounting Units of any Backer (assuming that there is a
market being made between the starting accounting unit and
the ending accounting unit). There are additional details
on the structure of these three types of contract that are
shown in Fig. 10-15.

25

Fig. 10 shows a simple Owner Change Contract 100 in
more detail. The Owner change is the simplest and most
common type of contract. The Holder remains the same as does
the Backer (i.e., the Accounting Unit). In this example,
30 there is only one Holder 58 who is also the Executor 64 and

one Backer 62, since the fee is being paid in the same Accounting Unit as the transfer.

However, there are two Owners 60a and 60b; otherwise
5 there could be no change of Owner. All accounts must be on the same Holder server, which must also be the Executor.

The only clauses allowed in the Clause section are the Fee Clause 90 and the Owner Clauses 108. The Fee Clause
10 transfers 0.50 units of Backer 62 to the executor 58 in return for attempting to execute the signed contract. The first owner clause transfers 20.00 units of unit 62 to owner 60b; the MinRate of 1.00 means that the entire amount must be transferred and none can be taken by the executor as a
15 fee (for this contract, the fee is corrected by the 2.00 units in the Fee clause). The second owner clause is a zero-value clause, included so that the second owner 60b will appear as a debited account and will need to sign the contract before it is valid. The second signature ensures
20 that 60b agrees with the Agreement section that follows.

The Agreement section spells out what Owner 60b agrees to do for 60a in return for the 20 units of 62. The Signature section must include signatures by 60a and 60b,
25 since both appear as owners in the source portion of clauses.

Fig. 11 shows a more complex multi-party Owner change contract 100. Although there can only be one Holder/Executor
30 in an Owner Change contract, the number of owners, backers and Owner clauses is unlimited. This makes it possible to create multi-party swaps. None of the owner clauses will be

executed unless all of them are executed, i.e., if one party does not have enough units to complete their part in the exchange, none of the exchanges occur.

5 In this example there are three owners and three accounting units (backers). For purposes of example, the three accounting units are called by the common names Lumberbucks, SoftFrancs and JackpotChips and the three owners are called Bill, Sam and June. The contract says that
10 Bill gives Sam 200 Lumberbucks, Sam gives June 500 SoftFrancs and June gives Bill 175 JackpotChips. The executor/holder 58 takes 2% of each transfer as a fee for holding the balances and executing the trade. All amounts must be moved to the Executor's server before the trade is
15 initiated. And all three owners must sign the contract before it is executable.

 If Fig. 11 were modified to be a two-party Owner change
20 instead of three parties, it would then be similar to a Backer change 102. The differences are that in an explicit Backer change the Owner does not know who provides the other accounting units and may specify a limit price as opposed to a fixed price. So a two-party Owner change contract is like
25 an off-market, non-anonymous block trade and the Backer change contract is like an automatic buy or sell order with a range of amounts and prices that are possible to complete the order.

30 When a contract is being authored, signed and executed, it is necessary to know the Identities of each party to the contract. This is necessary for at least three reasons:

1. to know whether we wish to deal with the party (are they certified by someone we trust),
2. to know how to communicate with the party (using their Public Key and Notify Method, and
3. to learn if a party offers the service of executing a particular type of contract (i.e., conversion from one specific accounting unit to another).

The party authoring a contract agrees to provide Identity information about all the parties to the contract to all other parties to the contract. The author could obtain the Identity information from web pages, email messages, floppy disks, or any other medium. The method by which a party passes on Identity information to other parties is known as the SAXAS Name Service.

The SAXAS Name Service is implemented as Owner Change Contracts. Any party can and will act as a SAXAS Name Service at various times. The party needing the Identity information authors a contract requesting Name Service on a specific Public Key; these details are stored in the agreement section of the contract. The contract is sent to the Name Service party for execution. The Name Service inserts the desired Identity into the memo field and signs it. Any party can also pass their own Identity on to other parties by including it in their memo section of any contract and signing it.

Fig. 12 shows a Holder Change Contract 98 in more detail. There are two basic formats: from a Holder to the Backer, or vice versa. This is an example of the first

format: move 400 units backed by party 62 and owned by party 60 from holder 58 to backer 62. A fee clause is used to collect 1.15 units of 62 for performing the transfer. The Backer is always the Executor of holder change contracts and always receives the fee, if any. The Agreement section is empty. The Signature section requires the signature of the Holder 58, Owner 60 and Backer 62.

Fig 13 shows a Holder change contract 98 that goes from the backer to a holder. In this example, the Backer 62 is acting as the original Holder as well. The amount is moved to the party 58 as the new Holder. The Fee clause allows the executor, who is the backer in this case, to take up to 0.75 units and move it to the account described as 62-62-62 in HOB format (this is account 92k on Fig. 6, the same account that usually has a negative total for all the units issued by this backer. So in effect the backer is withdrawing some units from circulation by collecting this fee.

The holder clause transfers 400 units from the backer's server 62 to the server of party 58, making 58 the new holder. There can only be one holder clause per holder contract. This clause also allows the executor to take up to 0.2% of the amount as a fee. When a Holder to Holder change is executed as two contracts and uses the Backer as an intermediary, there is no information left on the original Holder as to the final destination of the amount. Only the Owner and the Backer know that the funds moved from Holder A to Holder B.

Fig. 14 shows the Backer Change Contract 102 in more

detail. In this case, the Fee clause is in a different accounting unit from either of the units involved in the Backer Clause itself. This is permissible under the SAXAS architecture. The Backer clause 110 offers to give up 400 units backed by 60b in return for some units backed by 60c as long as the conversion MinRate is 0.552 or better. When the actual conversion is performed, the owner 60 will not be aware of who traded him the units. The executor matches up this contract with complementary contracts that offer to go the other way and also match in price. The actual conversion rate will be based on the offers available and a fee charged by the executor for this clause.

SAXAS allows for a single conversion clause to remain open and be matched up with multiple complimentary conversion clauses created by other owners. The clause will remain active and available for matching until the full amount offered for conversion has been converted, or the owner is found to have insufficient funds in his account, or the expiration date occurs. The details of each partial conversion are stored in the executor's memo field of the contract, and re-signed by the executor after each partial conversion. See Fig. 18-21 for operational details.

Operation of the SAXAS System. Figures 15-21

Each SAXAS account is uniquely defined by an identity triplet: Holder, Owner and Backer. It is possible for each component of the triplet to be a different party, the same party, or a mixture of parties.

The Holder is the party where the account resides. The Owner is the party who signs for the units and controls the account. And the Backer is the party who defines the accounting unit for the account and "backs it". Every party
5 in the system has a SAXAS account server and thus has accounts where it is the "holder", although these may exist only to balance with accounts that the party has with other parties.

10 Contracts all have an "Executor" that signs that a contract has been executed so that other parties to the contract know to update their books on notification. The executor is the final authority, or "commit coordinator". In the case of owner change or backer change contracts it is
15 the holder of the account. In the case of holder change contracts it is the backer of the currency, or central bank for that currency.

The executor does not execute a contract unless all
20 owners of accounts giving up something (and an owner change can have more than one) have signed. Also, in a holder change the holders have to sign. The holder where the units are going to has to agree to hold the units. The holder where the units are coming from has to agree that the owner
25 really had that much.

Fig. 15 shows the same sample set of accounts as in Fig. 6, but in this case the element numbers have been replaced with friendly names to make the logic easier to
30 follow. Assume that the friendly name of the party backing the accounting unit is

\$-Backer, the names of the parties acting as holders are H-Bank and Union Bank, and the names of the parties acting as Owners are Bob, Alice and Ted. Each Party has some Accounts on their server and some accounts on other servers, as shown in Fig. 6. It is noted that any accounts residing on a party's server will have that party as the "Holder" of the account, even though that party is acting as an Owner in the overall SAXAS scheme.

Owner Change.

Fig. 16 shows the execution of an owner change contract. See Fig. 10-11 for the structure of an owner change.

Processing is simplified because the owners must move all amounts to be swapped onto the Executor's server. No other servers are communicated with in the execution of a contract, although the server does send a notification copy of the completed contract to each party so that they can update their book keeping.

The optional fee clause is handled separately from the other owner change clauses. It is executed as soon as the executor receives a copy that is signed by the author and the fee payor, regardless of whether all the clauses of the contract turn out to be executable or not.

The executor verifies that any parties appearing as "debited owners" in the contract have properly signed the contract. Execution of an Owner Change is done as a unit, regardless of how many clauses and owners are involved. The

Executor logically executes all clauses simultaneously. This is done by locking all the balances in the database, executing all the transfers, then canceling the entire operation if any of the accounts falls below the required
5 Minimum Balance (usually 0.00 or the largest negative number).

This execution model means that you can exchange something in an Owner change that you do not own at the
10 start of the execution, but acquire during execution of the clauses. The Executor computes any percentage fees to be charged on each clause, but does not actually credit them until the end. Otherwise the Executor would have to lock the Executor's account for the entire execution time of the
15 contract and this would make the process inherently single threaded, since the Executor is involved in every contract on the server.

As a final step, the Executor attempts to send a
20 notification copy of the contract as executed to each party. This may involve use of the SAXAS Name Information Service to discover where the parties are currently to be notified. Notification can be via email if a party is not currently online, or notification may be optional.

25 Returning to Fig. 15 with the friendly names, assume that there is a SAXAS Owner Change contract (as in Fig. 10 and 16) with a clause transferring \$3 from Bob to Alice, the contract executing on the H-Bank server where they both have
30 accounts. Since each account is an (HOB) "triplet", the overall transfer as executed by the H-Bank server can be summarized as

(H-Bank,Bob,\$) \$3→ (H-Bank,Alice,\$)

So on the executor server (H-Bank) there are 2 accounts

5 updated:

(H-Bank,Bob,\$) -\$3 H-Bank is holding 3 less for Bob
 (H-Bank,Alice,\$) +\$3 H-Bank is holding 3 more for Alice

10 However, to keep the entire SAXAS system in balance,
 there are other accounts that must be changed as well.
 Starting with the H-Bank server, when it executes the
 contract, it updates the two accounts mentioned above. Then
 it sends a notification copy of the contract to Bob and to
 15 Alice, whose SAXAS servers must also update accounts that
 they hold:

On Bob's Server:

20 (Bob,Bob,\$) \$25 - \$3 = \$22 Total \$ held by Bob
 anywhere
 (Bob,H-Bank,\$) -\$5 + \$3 = -\$2 \$ held by H-Bank for Bob

On Alice's Server:

25 (Alice,Alice,\$) \$15+\$3 = \$18 Total \$ held by Alice
 anywhere
 (Alice,H-Bank,\$) -\$15-\$3 = -\$18 \$ held by H-Bank for Alice

30 Holder Change

Fig. 17 shows execution of a Holder Change Contract.
 See also Fig. 12-13 for the structure of Holder Changes.

A Holder Change Contract is like a wire transfer in the traditional world of financial transactions - an amount is moved from one Holder to another, but retains the same Owner and Accounting Unit (i.e. Backer).

The receiving holder has to agree (sign) to hold the units before the contract is executed. This allows a holder control over whom it holds units for by automatically verifying that the party's Identity contains certain "certificates". It also makes the holder promise that he is just holding the units and that it is really still the owners.

An honest holder will have enough units on record with the Backer to be able to cover all of the owners he is holding for. When units leaves one holder the Backer for that currency will refuse a contract if the holder does not have the units to cover it. A backer is like a central bank for clearing inter-holder transfers in his currency. And acting as the central bank for his currency, the backer is the executor for all holder changes in his currency.

There are three cases of holder change, although the first two are a necessary and sufficient core. The third is an alternate embodiment of the method.

- 1) You can transfer from a holder to the backer
- 2) From the backer to some holder
- 3) From one holder to another where neither is the backer

Case 1: The sequence of events when going from a holder to the backer is:

- 1) Owner signs holder change contract
- 5 2) Backer signs indicating "agree to hold for this owner"
- 3) The old holder
 - A) Verifies that the owner has the units
 - B) Updates his books and signs (he expects Backer to execute)
 - 10 C) Sends copy to both the backer and the owner
- 4) The backer executes the contract, updating his books, and change the contract status to "executed"
- 5) Backer notifies owner and old holder that contract is executed

15

Case2: The sequence of events going from the Backer to another holder is:

- 1) Owner signs holder change contract
- 2) Destination holder signs indicating "agree to hold
- 20 for this owner"
- 3) The backer signs and executes after checking the balance
- 4) The backer notifies owner and new holder that the contract is "executed" and they then update their
- 25 books

Case 3: Moving from holder A to holder B where neither is the backer (see Alternate Embodiments later in this document).

30

If an owner were moving units from one server to another, there is the danger that the first server had updated his books, but the new server did not receive the

notification and thus had not credited the owner with the units in his books. To correct this invalid situation, the owner would send the new holder another copy of the contract; if the contract is valid and has not been
5 processed on the destination server yet, the server will update its books to credit the owner with units.

The Holder Change contract allows the Owner to move value amounts from one Holder to another, but without
10 changing the Accounting Unit (Backer). The contract is signed by the Owner, the Holder and the Backer, to ensure that the amount will not be rejected when it arrives at its destination. The author will create two contracts, one to move the amount from the current holder to the backer and
15 the second to move it on to the new holder. The reason for this is that it makes each contract atomic and unambiguous. The owner can always verify where the funds are - they are never in limbo. If the funds get to the backer, but cannot be transferred to the new holder, they are still in the name
20 of the owner on the backer's server and the owner can verify their status.

An Example with 2 Holder Change Contracts

25 Returning to Fig. 16 with the friendly names, assume that two SAXAS holder change contracts are created, with the first transferring the amount to the Backer and the second transferring it to the final destination Holder. In both
30 cases owners creates the contracts and the \$-backer is the "executor".

Going back to the original state of accounts in Fig. 6a, assume that Bob wants to move \$5 from Union Bank to H-Bank. That can be represented in (HOB) triplet notation as these two transfers:

Contract 1: (Union, Bob, \$) \$5 → (\$, Bob, \$)
 Contract 2: (\$, Bob, \$) \$5 → (H-Bank, Bob, \$)

Contract 1: Holder to Backer

This "(Union, Bob, \$) \$5 → (\$, Bob, \$)" contract is NOT executed on the Union Bank server because it will be executed on the \$-Backer server. However, Union Bank must both ensure that Bob has sufficient funds and ensure that he does not spend those funds once the transaction is committed. The S-Backer server must verify that Union has enough total funds to perform the transfer (although the Backer does not know if Bob has enough funds). So both servers have to sign off on sufficient funds.

The way this is handled in SAXAS is that the original holder "tentatively transfers" the funds out of Bob's account and into the Backer's account in anticipation of the contract execution. This is a reasonable expectation because the executor-Backer has already signed the contract saying that he will perform it once the holder signs it. If for some reason the contract does NOT go through to execution, the holder will received as negative notification from the executor and must be prepared to move the funds back to Bob's account.

Tentative Transfers On the Union Server:

(Union,\$,\$) $-\$30 + \$5 = -\$25$ Total \$ held by Union
Bank

(Union,Bob\$) $\$20 - \$5 = \$15$ \$ held for Bob by Union
Bank

5

Then the Union Bank server sends a notification copy to
\$-Backer and to Bob.

On the \$-Backer Server:

10 (\$,Bob,\$) $\$0 + \$5 = \$5$ Open account for Bob,
deposit \$5

(\$,Union,\$) $\$30 - \$5 = \$25$ Reduce \$ holdings of Union
Bank

15

On Bob's Server:

(Bob,Union,\$) $-\$20 + \$5 = -\$15$ Reduce \$ held at Union
Bank

(Bob,\$,\$) $\$0 - \$5 = -\$5$ \$-Backer now owes Bob \$5

20

The executor also sends a notification copy to Union,
the original holder. However, since it has already does the
transfers in anticipation of this notification, it merely
marks the contract as complete.

25

Contract 2: Backer to Holder

The \$5 is now at the Backer's server. A second contract
is executed to transfer it to H-Bank, still in the name of
Owner Bob: "(\$,Bob,\$) $\$5 \rightarrow$ (H-Bank,Bob,\$)"

30

On the \$-Backer Server:

(\$,Bob,\$) $\$5 - \$5 = \$0$ Close out Bob, withdraw
\$5

(\$,H-Bank,\$) $\$20 + \$5 = \$25$ Increase \$ holdings of H-Bank

Then \$-Backer sends notification copies of the signed contract to H-Bank and to the Owner Bob, so that there book entries can be updated as well. Here are the transfers on each server:

On H-Banks Server:

(H-Bank,\$,\$) $-\$20 - \$5 = -\$25$ Total \$ held by H-Bank
 10 (H-Bank,Bob,\$) $\$5 + \$5 = \$10$ Bob \$ at H-Bank up by \$5

On Bob's Server:

(Bob,H-Bank,\$) $-\$5 - \$5 = -\$10$ Increase \$ owed Bob by H-Bank
 15 Bank
 (Bob,\$,\$) $-\$5 + \$5 = \$0$ Close out temp account at \$-Backer

20 Backer Change

The Backer Change Contract is a conversion of accounting units. The Owner and the Holder remain the same.

25 Fig. 18 shows execution of a Backer Change clause in a Backer Change Contract (see Fig. 14 for the structure). As with other contracts, the fee clause is executed at once when the contract is recorded by the Executor. There may
 30 also be percentage fees extracted from each conversion. The Executor can be any holder that makes a market. The conversion fees charged by that Executor are described in their identity record.

Each backer change clause of the contract is executed independently. Although the Backer change contract as described statically in Fig. 15 appears to involve only the Holder and the Owner. In fact the Holder matches the
5 contract with one or more reciprocal contracts by other Owners, contracts which offer the reverse conversion at a MinRate which allows matching of the contracts.

Each clause has a MinRate value, which is the minimum
10 rate at which the conversion can occur. On the initial execution of a new Backer Change clause, the actual rate at which it occurs is higher than MinRate, if the Executor can find a matching contract going the other way that allows for the Executor's fee. Partial conversion is supported, as well
15 as conversions by matching multiple reciprocal contracts.

After the initial execution, if there is any amount left to be converted, the contract goes into the outstanding offers list. If the contract is matched with a new incoming
20 contract before it expires, it receives exactly the MinRate.

The Backer Change Contract cannot be shown using Fig. 15, since Fig. 15 only shows one Backer for all amounts. Therefore, go back to the initial state of Fig. 15 and
25 assume a new party Y acting as Backer for a new accounting unit, CyberYen. Y authorizes Union Bank to hold accounts backed by Y and Union Bank offers a service converting Y units into \$ units and vice versa, all performed by the SAXAS server.

30

If Owner Ted wants to convert \$5 from his Union Bank account into Y at a minimum rate of 100Y per \$, this can be notated as a Backer Change Offer in SAXAS:

5 (Union, Ted, \$) \$5 @10 → (Union, Ted, Y)

If the Union SAXAS server can complete this Backer Change and sends the notification copy to Ted, the following transfers occur:

10

On Union Server

(Union, Ted, \$) \$10 - \$5 = \$5 Extract \$5 from Ted's account

15 (Ted, Union, \$) -\$10+\$5 = -\$5 Reduce \$ owed by Union to Ted

On Ted's Server

20 (Ted, Ted, \$) \$10 - \$5 = \$5 Reduce total \$ owned by Ted

(Ted, Union, \$) -\$10+\$5 = -\$5 Reduce \$ owed by Union to Ted

(Ted, Ted, Y) Y0 + Y500 = Y500 Total Y units owned by Ted

(Ted, U, Y) Y0 - Y500 = -Y500 Y units owed to Ted by Union

25 However, where did the Y500 come from and where did the \$5 go to? This Backer Change can only occur if there is someone else with a contract on the Union server offering to convert 500Y to \$ at a rate of 0.01 Y per \$. Assume that Bill is a party with Y1000 and \$0 and the following SAXAS contract offer. Then the Union server can match up Bill's contract with Ted's contract and make the conversion:

30

(Union,Bill,Y) Y500 @0.01 → (Union,Bill,\$)

On Union Server

(Union,Bill,Y) Y1000-Y500=Y500 Extract Y500 from Bill's

5 acct

(Union,Bill,\$) \$0 + \$5 = \$5 Credit new account \$5 for
Bill

On Bill's Server

10

(Bill,Bill,Y) Y1000-Y500=Y500 Reduce total Y owned by
Bill

(Bill,Union,Y) -Y1000+Y500=-Y500 Adjust Bill's Y at Union

(Bill,Bill,\$) \$0 + \$5 = \$5 Total \$ owned by Bill

15 (Bill,Union,\$) \$0 - \$5 = -\$5 \$ owed by Union to Bill

As a result of these Backer Changes, the total Y and \$
held by Union Bank has not changed, but they are allocated
to different Owner accounts. In order for the Union server
20 to stay in balance, both of these Backer Changes must be
executed at the same time as a single unit, even though
there are two contracts.

Fig. 19a shows an example of two reciprocal backer
25 change clauses that will be matched and executed. Below are
two contracts that can be matched up and executed, if the
fee is zero. Assume Holder A, Owners B and C, Backers D and
E:

30 ABD => ABE 100 @ MinRate 0.5 (100D * 0.5 = 50 E)
ACE => ACD 100 @ MinRate 2.0 (50E * 2.0 = 100 D)

This is interpreted as "Owner B offers to exchange up to 100 units of D for 0.50 units of E per D, or better. And Owner C offers to exchange up to 50 units of D, the price being 2.0 E per D or less". The executor can match these two
 5 contracts, exchanging 50 units of E from owner C for 100 units of D from owner B.

In the example of Fig. 19a, the reciprocal backer clauses are at exactly the same price. This does not leave
 10 any spread for the executor to take a fee.

Fig. 19b shows a different situation where there is a spread between the bid and offer.

15 ABD => ABE 100 @ MinRate 0.48 100D * 0.48 = 48 E
 ACE => ACD 100 @ MinRate 2.00 50E * 2.00 = 100 D

The Fig. 19b example is the same as Fig. 19a, except that owner B has lowered his price. He will part with 100 D
 20 units for 48 E units instead of 50 E. Since owner C will give up 50 E units for 100 D, there are 2.0 units of E left over, from which the executor can extract his fee.

Assume that the executor fee in Fig. 19b is only 1.0 E.
 25 The remaining 1.0E of spread goes to the new contract. If we assume that B's contract is new and C's is already in the system waiting for new offers, B would get 49E for 100D instead of 48E.

The fee that the Executor will charge must be stored in the
 30 database somewhere so the software can match the trades automatically. They are available to the outside world via a Name Service request.

Fig. 20-21 shows waiting bids and offers and how they are matched with a new, incoming contract. Note that none of the existing Backer change clauses can be matched because
5 the sellers are asking more than the buyers are willing to pay. So the clauses wait on the Holder's server for a new Backer change clause to appear.

If the new contract falls in the price gap between the
10 unmatched clauses, it will be matched with one or more of them.

The new offer to sell 60 units of D at a MinRate of 4.25 E per D can be matched with the two best offers to sell
15 D for E.

The best waiting offer is to sell 50E at 4.40 E per D, which equals 11.37D, plus a 1% commission of 0.1137D, gives a price of 11.4837D.

- 20 • 11.37D goes to the party selling the 50E.
- 50E goes into the account of the new contract.
- 0.1137D goes to the executor.
- 11.4837D comes out of the account of the new contract.
- This leaves an outstanding new contract to sell
25 48.5163D at 4.25E or better.
- The effective rate on this exchange was 4.35 E per D to the new contract, or better than the offer of 4.25.

The second best waiting offer is to sell 137E at 4.30 E
30 per D, which matches the amount remaining on the new contract. Therefore this order is matched as well,

generating a price of 31.86D plus a 1% commission of 0.3186D, which equals 32.1786D.

- 31.86D goes to the party selling the 137E.
- 5 • 137E goes into the account of the new contract.
- 0.3186D goes to the Executor.
- 32.1786D comes from the account of the new contract.
- The effective rate for this exchange is 4.257, which is better than 4.25.
- 10 • There is still an outstanding offer to sell 16.3377D at 4.25 E per D, but there are no more waiting offers to match.

Therefore, the outstanding balance goes into the list
15 of buy and sell orders as the best unmatched sell offer. It will be matched with the first new contract that comes in and offers 4.25. The new contract gets the benefit of the best price. The old contracts that are waiting on the server for further matches will always get their MinRate and no
20 better, but they don't have to pay any fees. The new contract effectively pays the Executor's fee if there is one.

Fig. 21 shows the resulting buy and sell orders after
25 the new contract is processed. Notice the two offers to sell E for D are gone and new offer to sell D for E has been added.

Alternative EmbodimentsAlternative Execution Models

5 In the preferred embodiment, a clause in a SAXAS
contract changes only one element of the Identity triplet
defining an account: either the holder, the owner, or the
backer. This simplifies the design and the implementation
and makes it less likely that any units will ever be lost.
It is also desirable to change two of the elements in a
10 single clause.

Since any arbitrarily complex transfer can be
accomplished with combinations of contracts, other
embodiments of the SAXAS user interface can allow an owner
15 to transfer an amount from any currency, any place, to any
other owner in any other currency at any other place, using
the contracts in the preferred embodiment.

Also, the holder change can be performed in a single
20 contract instead of two contracts (holder to backer and
backer to holder). The logic is similar to the two contract
method, but occurs as a single logical step:

- 1) Owner signs holder change contract
- 25 2) Holder B agrees to hold for that owner
- 3) Backer agrees to hold for the holder B
- 4) Holder A checks that owner has the funds and updates
his books
- 5) Backer checks that holder A has funds, updates books
30 and marks contract as "executed" and notifies owner, B,
and A

6) Owner, and holder B update their books

The Backer is the executor of Holder Change contracts, because the Backer is the only one who can give any
5 assurance that the holder still has the amount that is being transferred. Actually, the Backer can only verify that the holder has at least the amount in the contract in total funds; the Backer cannot verify that the holder has not done something amiss with other amounts.

10

Alternate Software Methods

The preferred embodiment of this invention was written in the Java language, but there is no reason why it could not be rewritten in C, C++, COBOL, FORTRAN, Pascal,
15 CodeWarrior for the Palm Pilot, or any other computer language.

Alternate Data Store Methods

The preferred embodiment of this invention used the
20 JDBC (Java Database Connectivity) interface to databases, but the invention could be implemented with any database interface that supports tables and transaction commit, including but not limited to DB-2, Oracle, Sybase, Mini-SQL.

Alternate Packaging

25 The preferred embodiment of this invention is a standalone program that performs its own communication and display, but it could also be embodied as a browser plug in.

Alternate Computing Devices

The preferred embodiment of this invention is an Intel-Microsoft personal computer, but the invention can be easily implemented on a wide range of computing platforms, including but not limited to UNIX, IBM AS/400, Hewlett-Packard MPE/iX, DEC VAX, and IBM mainframe. A database is not needed; only a way to store and recall contracts. A network connection is not needed, only a way to send a contract from one party to another. The contract could even be printed, mailed, scanned, converted back to characters by optical character recognition, and it would still be valid.

A computer is not strictly needed, only an algorithmic device that can parse the contract, compute the mathematical signatures, and send the contract on in some fashion. Given advances in various technologies, this could be a specialized piece of hardware or a genetically engineered biological entity.

While the invention has been described with reference to specific embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted for elements thereof without departing from the true spirit and scope of the invention. In addition, modifications may be made without departing from the essential teachings of the invention.

We claim:

1. A method for exchanging ownership of values among two or more parties, comprising the steps of

providing a computing device as a means to input, output and store information and perform mathematical and logical algorithms on said information,

providing a secure channel for sending information intended for a specific party from said computing device to another computing device, and

providing a computer program to control said computing device, said computer program being capable of:

operating on several computing devices under control of native parties that control the computing device,

creating a digital identity for said party, said digital identity comprising a unique, verifiable sequence of bits that can create a digital signature for a specific digital document and a plurality of optional attributes, said optional attributes comprising the specific secure channel to be used in sending information to the party, the real world identity of the party, and the exchange services provided by the party,

receiving said identities when sent over said secure channel,

creating an accounting unit for said value, such accounting unit being divisible into fractional parts, being transferable, and being defined by a backer party who defines terms under which said accounting unit may be surrendered for said value,

receiving information regarding said accounting units when sent over said secure channel,

creating accounts, said accounts to contain a balance amount, said balance amount to be recorded in one of said

accounting units, controlled by a owner party, and entrusted to a holder party, and

creating a contract in digital form.

1 2. The method for exchanging ownership of values among two
2 or more parties as set forth in Claim 1, further including
3 receiving said contracts when sent over said secure
4 channel, and
5 sending said contracts over said secure channel to
6 parties on other said computing devices.

3. The method for exchanging ownership of values among two
or more parties as set forth in Claim 2, further including
signing said contract using said digital identity of a
party native to said computing device,
5 verifying that said contract is validly signed by the
digital identities that are named as parties in said
contract,
executing said exchange clauses in said contracts when
said contracts are validly signed by all impacted parties
10 and when the designated executor party is native to said
computing device.

4. The method for exchanging ownership of values
among two or more parties as set forth in Claim 3, wherein
said step of executing comprises the steps of
moving accounting units from one owner party to another
5 owner party,
converting amounts from one accounting unit to another
accounting unit,

matching and executing reciprocal buy and sell
contracts, transferring amounts from one holder party to
10 another holder party via the control of the backer party,
adjusting account balances in a double-entry manner so
that they balance to zero in total,
optionally processing said textual agreement, causing
information on said parties, said accounts, or said
15 contracts to be appended to the contract and signed by the
executor,
notifying all parties to the contract by sending a copy
over said secure channel, and
optionally signalling software programs external to the
20 method that said contract has been executed.

5. The method for exchanging ownership of values among two
or more parties as set forth in Claim 4, wherein said
contract comprises

a header, said header comprising the author party and
5 executor party and contract type, said contract types
specifying special interpretation of the text agreement,
a list of party identities,
a list of exchange clauses,
said text agreement comprising a legal agreement among
10 the parties and, optionally, directions to be interpreted by
the software program for specific contract types, and
a list of digital signatures by said parties.

6. The method for exchanging ownership of values among two
or more parties as set forth in Claim 5 further including
creating said exchange clauses, said exchange clauses
comprising

5 a route for the value exchange, said route comprising a
from account and a to account,
an amount,
a debit amount indicator if said amount is to be
removed from said from account or a credit amount indicator
10 if said amount is to be added to said to account,
a minimum conversion rate, and
a mortality field to determine how long the clause
should remain executable.

7. A secure architecture for performing and managing value
exchanges among parties comprising:
a formalized legal contract in a special format,
a public key that identifies each party to the value
5 exchange,
a software program for use on a programmable computer
for each party that can create, validate, sign, store,
encrypt, transmit, and execute contracts,
a communication link between said computers for
10 transmitting the value exchange information, and
a database for each party to hold the contracts and
identify information for each party.

1 8. The secure architecture of Claim 7 wherein said public
2 key is also used to encrypt the transmitted value exchange
3 information to ensure that only the intended party can read
4 the message transmitted.

1 9. The secure architecture of Claim 8 wherein each party
2 is identified by a unique identity, said identity including
3 said public key.

1 10. The secure architecture of Claim 9 wherein said unique
2 identity includes at least one of a party's preferred name,
3 telephone number, Internet web page address, and system
4 services offered.

1 11. The secure architecture of Claim 7 further including a
2 personal computer upon which said software program is
3 installed.

1 12. The secure architecture of Claim 11 wherein said
2 personal computer's operating system is Windows 95 or
3 Windows 98.

1 13. The secure architecture of Claim 12 wherein said
2 operating system also include the JAVA program.

1 14. The secure architecture of Claim 7 wherein said public
2 key comprises a sequence of unforgeable characters that
3 uniquely and securely identify a particular party.

1 15. The secure architecture of Claim 14 wherein said public
2 key further includes a digital certificate, said certificate
3 comprising the real world identity of a particular party,
4 signed and verified by a certifying authority.

1 16. The secure architecture of Claim 7 wherein each party
2 to the transaction has an account, said account comprising
3 the identities of the parties, said parties including at
4 least a Holder, an Owner, and a Backer.

1 17. The secure architecture of Claim 16, wherein a Holder
2 is one of a bank, an escrow agent, or a broker.

1 18. The secure architecture of Claim 16, wherein an Owner
2 is the signatory for one or more accounting unit balances,
3 said accounting unit comprising one of any legal tender, or
4 other unit of value for which one party would promise to
5 redeem in return for something with a real world value.

1 19. The secure architecture of Claim 16, wherein a Backer
2 is that party to the transaction that issues an accounting
3 unit, said accounting unit comprising one of any legal
4 tender, or other unit of value for which one party would
5 promise to redeem in return for something with a real world
6 value.

20. The secure architecture of Claim 7, wherein said
formalized contract comprises at least four electrically
signed sections, said sections comprising

5 a header section which includes at least one of a
title, author, executor, date of contract maturity,

a parties section which includes the identification of
the party,

a clauses section which includes the amount of the
contract and the interest rate,

10 an agreement section which includes the actual text of
the contract agreement.

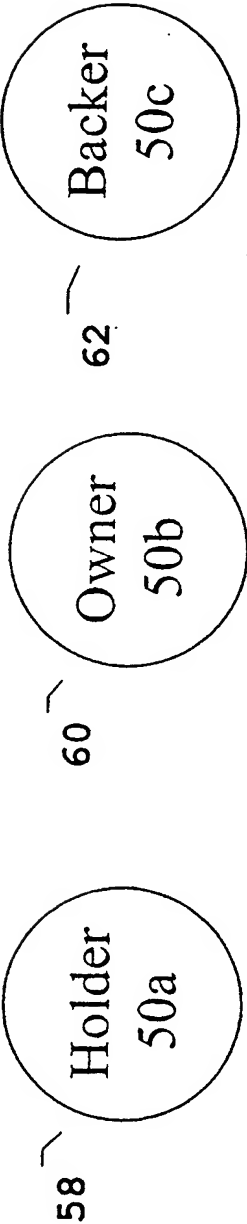
1 21. The secure architecture of Claim 20, further including
2 a fifth section, said section comprising the electronic
3 signatures of all the parties to the contract.

1 22. The secure architecture of Claim 12 wherein said
2 operating system also includes at least one of the following
3 programs: C, C++, COBOL, FORTRAN, Pascal, CodeWarrior.

1 23. The secure architecture of Claim 7 further including a
2 database interface, said database interface including at
3 least one of DB-2, Oracle, Sybase, Mini-SQL.

1 24. The secure architecture of Claim 11 wherein said
2 personal computer is one of the following computers: Intel-
3 Microsoft family of computers, UNIX, IBM AS/400, Hewlett-
4 Packard MPE/iX, DEC VAX, IBM mainframe.

Parties Play Roles in Exchanges



and Own Resources that Create a Server

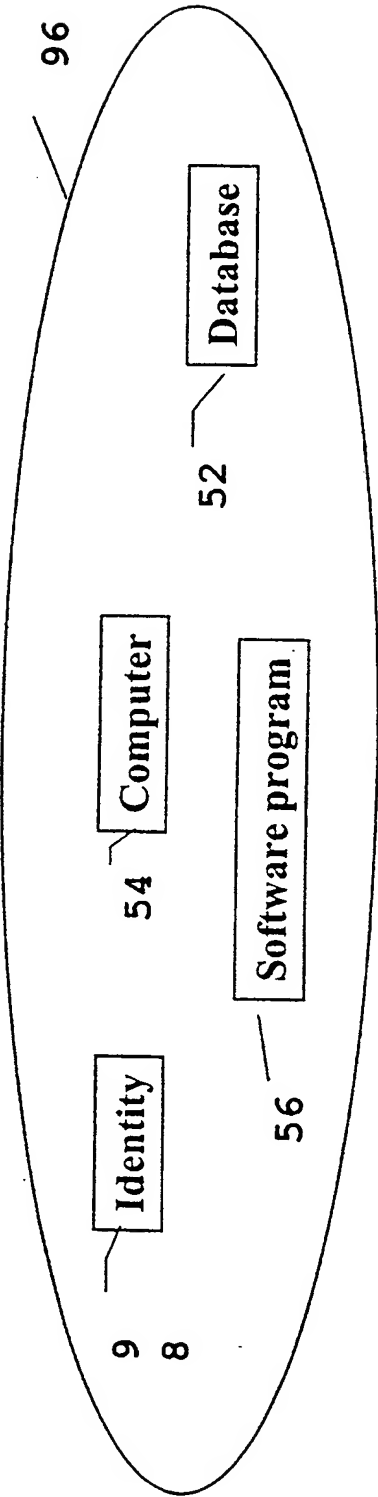


Fig. 1

A Public Key

```

-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCpDAu9ZARL2pzeRxtYzKe5LDevPBXx6fV77CjK
rZXgHFqKWY9Ab9UywfNDNL4rF5EDRPqdoyXPuns2SktQj58QnWwY7t+vCpIQJbPudsA8Ls1gr7Xu
FOTEmh0TRCifeHXhXTdqsR71k80KwUJsSFyoEIx4Iozx1B3GRdpvGOp6zwIDAQAB
-----END PUBLIC KEY-----

```

68

A Digital Signature

```

-----BEGIN SIGNATURE-----
A1zbqXTv+7HvpEpazmngV/+TLLPVwbRi7wyhHB7+AmEvVzd4KD1u+V2erYyMglZW4tG2/HbvImWN
4kUOhKLr/SJeOYG24RG8mozTry19+exlfBQmxjEElek+slfXPUjRTmGoprZKFithqF4Sxc8xUU7+
NCLTgfIHivdkFXS89sh
-----END SIGNATURE-----

```

114

Fig. 2

X.509 Certificate

```

-----BEGIN X509 CERTIFICATE-----
MIICoTCCAgOAgFFMAOGCSqGSIb3DQEBAUAMIHEMQswCQYDVQQGEwJaQTEVMBMG
A1UECBMMV2VzdGVybiBDYXB1MRlWEAYDVQQHEw1DYXB1IFRvd24xHTABBgNVBAoT
FFRoYXN0ZSBDb25zdWx0aW5nIGNjMSGwJgYDVQQLEx9DZXJ0aWZpY2F0aW9uIFN1
cnZpY2VzIERpdmlzaW9uMRkwFwYDVQQDExBUaGF3dGUgU2VydMvYIENBMSYwJAYJ
KoZihvcNAQkBFhdzZXJ2ZXItY2VydHNAAdGhhd3R1LmNvbTAeFw05NzAxMjkyMTA2
MDhaFw05ODAxMjkyMTA2MDhaMGwxCzAJBgNVBAYTAkFJMQ8wDQYDVQQIEwZPbGQg
VGExKzApBgNVBAoTik9mZnNob3JlIEluZm9ybW9uIFN1cnZpY2VzIEw0ZC4x
HzAdBgNVBAMTFm9ubGluZS5vZmZzaG9yZS5jb20uYWkwZ8wDQYJKoZIhvcNAQEB
BQADgY0AMIGJAoGBAPSi1stCH2aS9zlawOQ1WxjoStzJQ66KDJCsoXQaywTOOFKO
DL6tOHO0cIsgjVdqfS1RgPCV9fGHuDuxDBzHNKmxR9VZZdzpfg/AxKkedMCy5gde
W0uDSq4wN2HzAT1Kntqeb8KhtfKqk7oltJxduX7QpFIDSg2k6Jxr3th6VS6pAgMB
AAEwDQYJKoZIhvcNAQEEBQADgYEApKkcq7v3+Aeg+ICtOlPmiRel1GQAsaTLe0/j
Hn1Mpqz2j5IL93nm6L800Ay6Yccz+hC6arQLLOCLG40IYf45yn197I4TwEXNnk2d
RcyzkZHYZg8dkUfp3M+chSRBm/d4qSxA+MT4sloes8pdJODUFMXElcgV8/A/wyhd
pIIgCA=
-----END X509 CERTIFICATE-----

```

Fig. 3

An Account

Holder	Owner	Backer	Balance
50a	50b	50c	+100

Fig. 4

Types of Accounts

Holder	Owner	Backer	Balance
50a	50b	50c	100
50b	50a	50c	-100
50a	50c	50c	-100
50b	50b	50c	100

Fig. 5

Balancing 1 Backer, 2 Holders, and 3 Owners

	92k	92l	92m	92n	92o	92p	92a	92q	92r	92s	92b	92t	92u	92v	92c	92w	92x	92y	92z
Holder 58	50c	50c	50c	50a	50a	50a	50b	50b	50b	50b	50b	50d	50d	50d	50d	50e	50e	50f	50f
Owner 60	50c	50a	50b	50c	50d	50e	50a	50c	50d	50f	50b	50d	50a	50b	50d	50e	50a	50f	50b
Backer 62	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c	50c

Fig. 6

Balancing When a Holder is Also An Owner

	92a	92b	92c	92d	92e	92f	92g
Holder 58	50c	50c	50a	50a	50a	50b	50b
Owner 60	50c	50a	50c	50a	50b	50b	50a
Backer 62	50c	50c	50c	50c	50c	50c	50c
Balance 94	-50	50	-50	10	40	40	-40

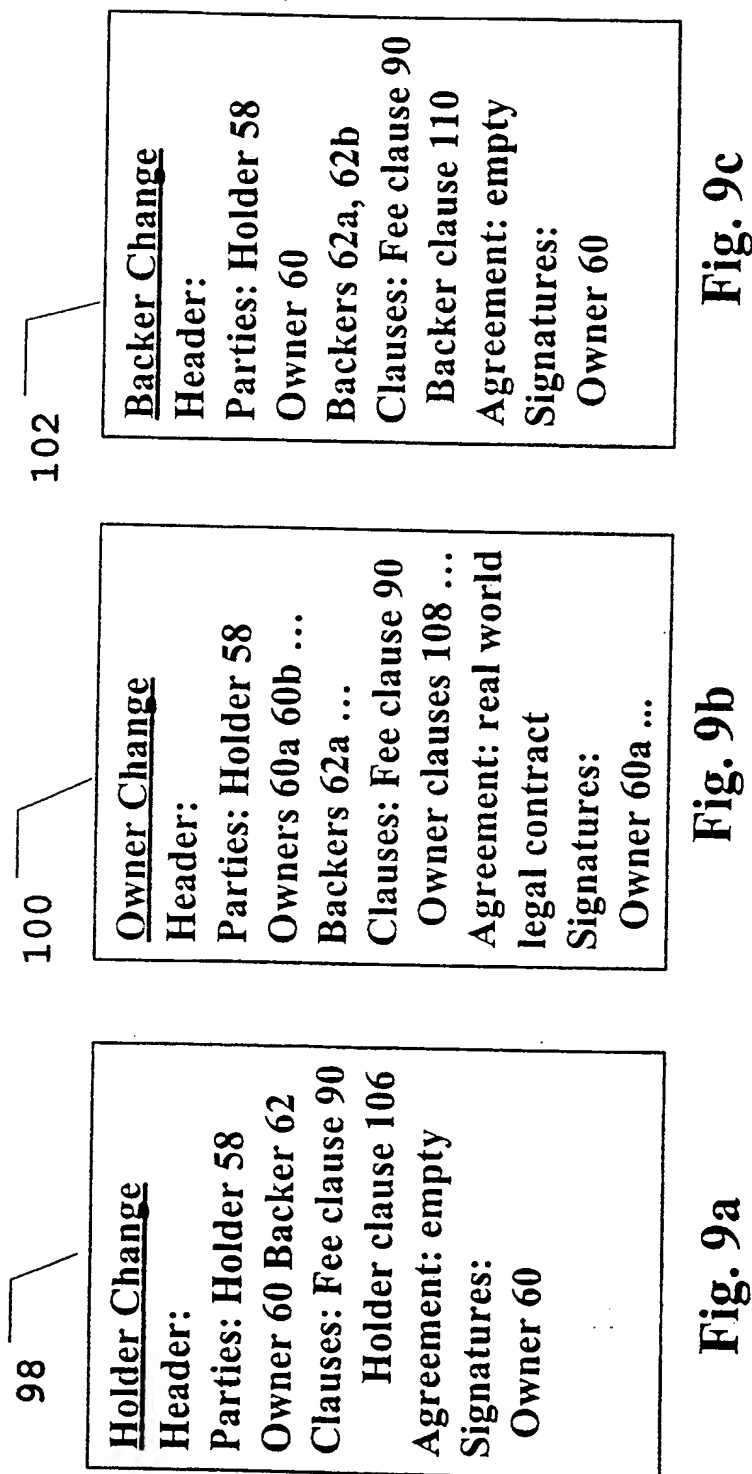
Fig. 7

Structure of a Contract - 66

Header - consisting of Title, Author, Executor 64, Arbitrator, When Contract Becomes Valid, When Contract Expires	78
Parties	80
- Friendly Name and Public Key 98 for each	
Clauses	82
- Debit HOB, Credit HOB, Amount, MinRate	
Agreement - Real World agreement	84
Signatures, consisting of Memo 88, Sig-Time, Status (rejected, I agree, Executed), Sig-Text	86

Fig. 8

Three Types of Contract



Single-Party Owner Change (with Agreement)

Header: Executor = Holder 58

Parties:

Holder 58 Owners 60a 60b Backer 62

Clauses:

Fee clause 90:

From	To:	Amount
58 60a 62	58 58 62	0.50

Owner clauses 108:

From	To:	Amount
58 60a 62	58 60b 62	20.00

From	To:	Amount
58 60b 62	58 60c 62	0.00

Agreement: Party 60b agrees to wash the car of 60a.

Signatures: Owner 60a, Owner 60b

Fig. 10

Multi-Party Owner Change

Header: Executor = Holder 58

Parties:

Holder 58 Owners 60a 60b 60c Backer 62a 62b 62c

Clauses:

Owner clauses 108:

From	To:	Amount
58 60a 62a	58 60b 62a	200

From	To:	Amount
58 60b 62b	58 60c 62b	500

From	To:	Amount
58 60c 62c	58 60a 62c	175

Agreement: This is a three-way trade in three accounting units.

Signatures: Owner 60a 60b 60c

100

Fig. 11

Holder Change Contract (From Holder to Backer)

Header: Executor = Holder 58

Parties:

Holder 58 Owner 60 Backer 62

Clauses:

Fee clause 90:

From	To:	Amount
58 60 62	58 58 62	1.15

Holder clause 106:

From	To:	Amount
58 60 62	62 60 62	400

Agreement: empty.

Signatures: Owner 60

Fig. 12

Holder Change Contract (From Backer to Holder)

Header: Executor = Backer 62

Parties:

New Holder 58 Owner 60 Backer 62

Clauses:

Fee clause 90:

From	To:	Amount
62 60 62	62 62 62	0.75

Holder clause 106:

From	To:	Amount
62 60 62	58 60 62	400

Agreement: empty.

Signatures: Owner 60

Fig. 13

Backer Change Contract

Header: Executor = Holder 58

Parties:

Holder 58 Owner 60 Backers 62a 62b 62c

Clauses:

Fee clause 90:

From	To:	Amount
58 60 62a	58 58 62a	0.60

Backer Clause 110:

From	To:	Amount	MinRate
58 60 62b	58 60 62c	400	0.552

Agreement: empty.

Signatures: Owner 60

Fig. 14

Friendly Names for 1 Backer, 2 Holders, and 3 Owners

	\$-Backer				H-Bank				Union Bank				Bob				Alice		Ted	
Holder 58	\$	\$	\$		H	H	H	U	U	U	U	Bob	Bob	Bob	Bob	Al.	Al.	Ted	Ted	
Owner 60	\$	H	U		\$	Bob	Al	H	Bob	Ted	U	Bob	H	U	\$	Al	H	Ted	U	
Backer 62	\$	\$	\$		\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	
Balance 94	-50	20	30		-20	5	15	0	-30	20	10	0	25	-5	-20	0	15	-15	10	-10

Fig. 15

Execution of Owner Change

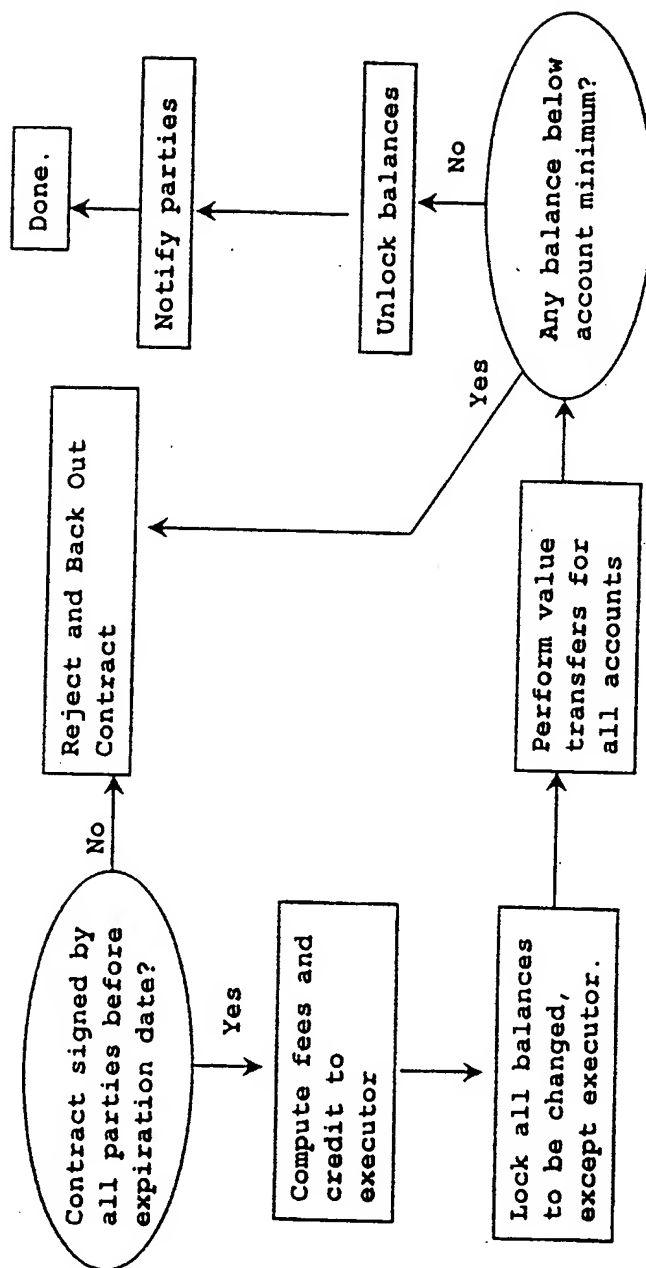


Fig. 16

Holder Change in 2 Contracts

WO 00/28452

17 / 21

PCT/US99/25853

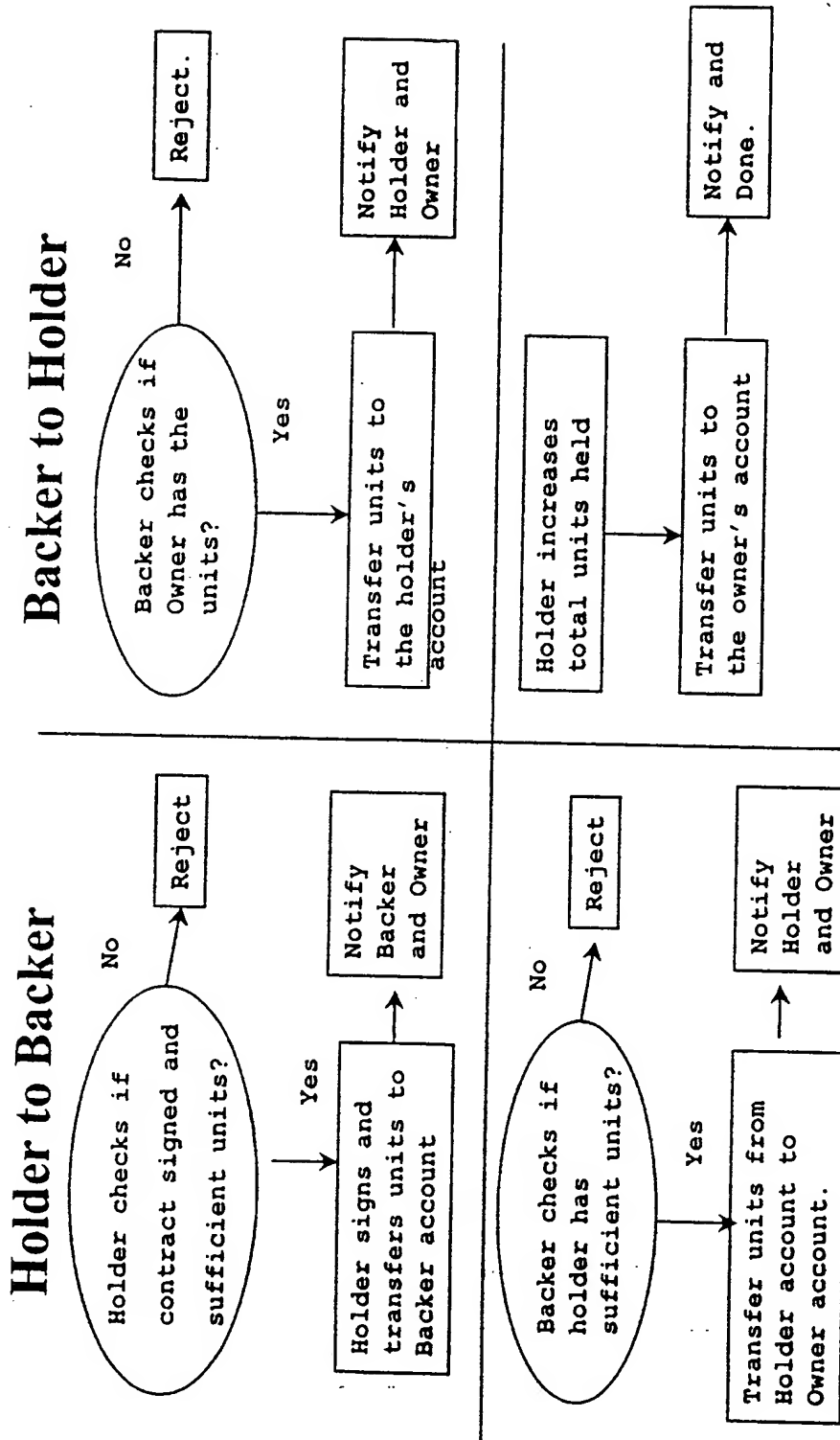


Fig. 17

Execution of a Backer Change

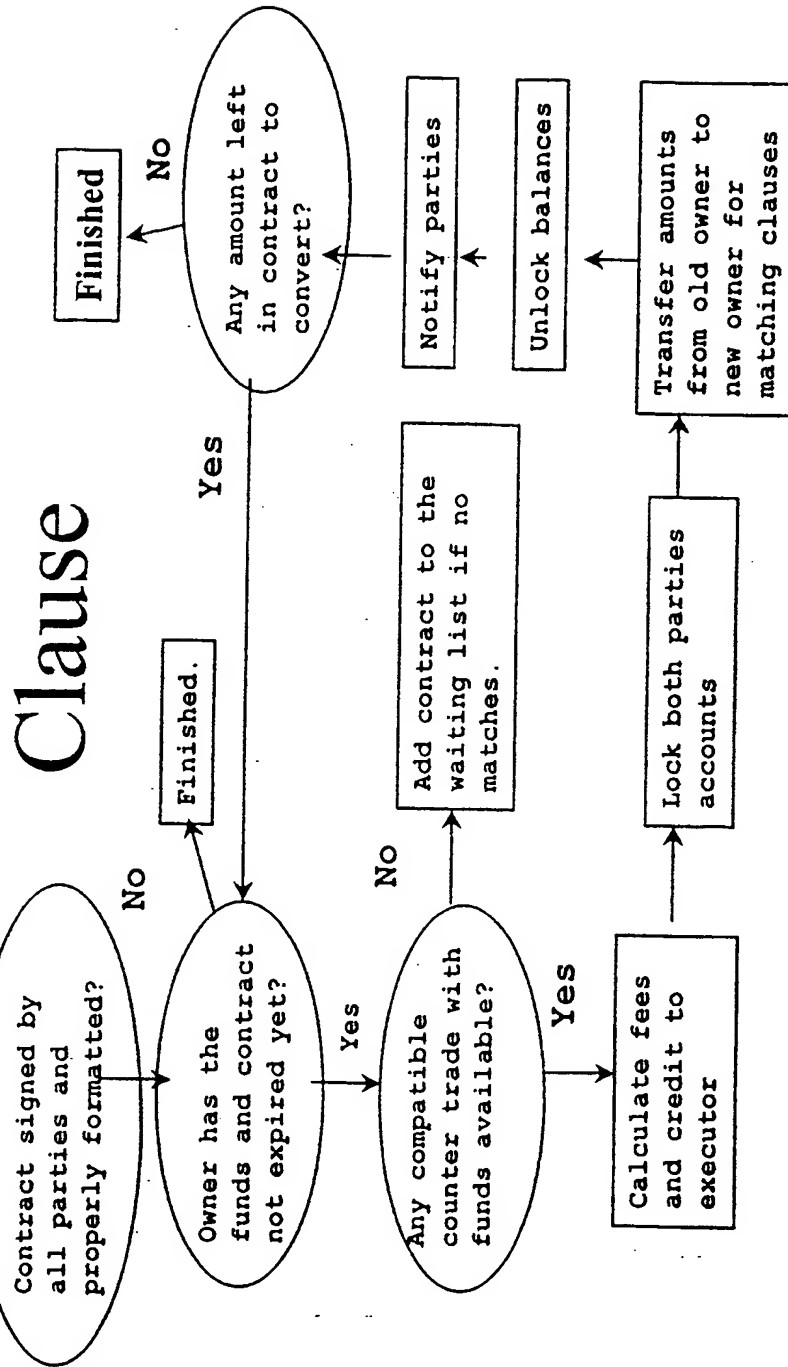


Fig. 18

Matching of Backer Changes

Assume Holder A, Owners B and C, Backers D and E:

ABD => ABE	100 @ MinRate 0.5	(100D * 0.5 = 50 E)
ACE => ACD	100 @ MinRate 2.0	(50E * 2.0 = 100 D)

Fig. 19a

ABD => ABE	100 @ MinRate 0.48	(100D * 0.48 = 48 E)
ACE => ACD	100 @ MinRate 2.00	(50E * 2.00 = 100 D)

Fig. 19b

Match New With Waiting Orders

Offers to Sell D for E

Sell 500D @ 4.70 E per D
 Sell 105D @ 4.60 E per D
 Sell 20D @ 4.50 E per D

New offer: convert 60 D into E @ MinRate= 4.25
 (1% commission)

Offers to Buy Sell E for D

Sell 50E @ 4.40 E per D (sold for 11.4837D)
 Sell 137E @ 4.30 E per D (sold for 32.1786D)
 Sell 20E @ 4.20 E per D (asking too much)

Fig. 20

After Matching New Backer Clause with Waiting Orders

Offers to Exchange D for E

Sell 500D @ 4.70 E per D

Sell 105D @ 4.60 E per D

Sell 20D @ 4.50 E per D

Sell 16.3377D @ 4.25 E per D

All orders matched, awaiting new contracts.

Offers to Buy Exchange E for D

Sell 20E @ 4.20 E per D (asking too much)

Fig. 21

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US99/25853

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 17/60; H04K 1/00; H04L 9/00

US CL : 705/77, 78, 80

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 705/77, 78, 80

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

BRS (secure transaction)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,757,917 A (ROSE et al.) 26 May 1998, column 2, lines 50-67, column 3, lines 1-67.	1-24
Y	US 5,826,241 A (STEIN et al.) 20 October 1998, column 3, lines 5-67, column 4, lines 1-67.	1-24
Y,P	US 5,870,473 A (BOESCH et al.) 09 February 1999, column 7, lines 18-67.	1-24



Further documents are listed in the continuation of Box C.



See patent family annex.

Special categories of cited documents:	
"A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E" earlier application or patent published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"I" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O" document referring to an oral disclosure, use, exhibition or other means	"A" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

21 February 2000 (21.02.2000)

Date of mailing of the international search report

17 APR 2000

Name and mailing address of the ISA/US

Commissioner of Patents and Trademarks

Box PCT

Washington, D.C. 20231

Facsimile No. (703)305-3230

Authorized officer

Gail O. Hayes

James R. Matthews

Telephone No. (703) 305-3900